

An enhanced cascade ensemble method for big data analysis

Ivan Izonin^{1,2}, Roman Muzyka², Roman Tkachenko³, Michal Gregus⁴, Roman Korzh⁵, Kyrylo Yemets²

¹Department of Civil Engineering, School of Engineering, University of Birmingham, Birmingham, United Kingdom

²Department of Artificial Intelligence, Lviv Polytechnic National University, Lviv, Ukraine

³Department of Publishing Information Technologies, Lviv Polytechnic National University, Lviv, Ukraine

⁴Faculty of Management, Comenius University Bratislava, Bratislava, Slovakia

⁵Department of Social Communication and Information Activities, Lviv Polytechnic National University, Lviv, Ukraine

Article Info

Article history:

Received July 8, 2024

Revised Oct 23, 2024

Accepted Nov 14, 2024

Keywords:

Big data analysis

Binary classification task

Cascade ensemble

Imbalanced dataset

Kolmogorov-Gabor polynomial

Machine learning

Wiener polynomial

ABSTRACT

In the digital age, the proliferation of data presents both challenges and opportunities, particularly in the realm of big data, which is characterized by its volume, velocity, and variety. Machine learning is a crucial technology for extracting insights from these vast datasets. Among machine learning methods, ensemble methods, and especially cascading ensembles, are highly effective for big data analysis. While it is true that the training procedures for cascade ensembles can be time-consuming and may have limitations in terms of accuracy, this paper proposes a solution to enhance their performance. Our method involves using stochastic gradient descent (SGD) classifiers, an improved training data separation algorithm, and integrating principal component analysis (PCA) at each ensemble level. We are confident that these enhancements lead to improved results and accuracy. The proposed approach is designed to enhance both the generalization properties and accuracy of the ensemble (3%), while also reducing its training time. Results from modelling on a real-world biomedical dataset demonstrate significant reductions in training duration, improvements in generalization properties, and enhanced accuracy when compared to other possible implementations of the ensemble.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Roman Muzyka

Department of Artificial Intelligence, Lviv Polytechnic National University

S. Bandera str., 12, Lviv, 79013, Ukraine

Email: roman.muzyka.mknssh.2022@lpnu.ua

1. INTRODUCTION

In the digital era, the exponential growth of data has ushered in new challenges and opportunities. Big data is often characterized by among other things, its volume, velocity, and variety, which can pose a formidable challenge to conventional data processing techniques. However, machine learning has emerged as a pivotal technology in addressing these challenges, due to its ability to analyze and extract insights from massive datasets. The synergy between machine learning and big data processing has undergone significant development over time. However, conventional data processing techniques have encountered challenges in handling the vast amount and intricacy of big data [1], [2]. The deployment of machine learning models at scale has been significantly improved by machine learning algorithms, particularly those that utilize parallel computing and distributed systems. This has allowed organizations to extract value from their data assets more efficiently [3]. The advent of technologies such as apache hadoop and spark has made scalable and efficient big data processing frameworks accessible, thereby facilitating the deployment of machine learning models at scale [4].

Machine learning has several advantages for processing big data tasks [5]. Firstly, it enables predictive analytics by identifying patterns and trends within vast datasets, which can facilitate informed decision-making [6]. Secondly, machine learning algorithms can automate data processing workflows, reducing manual

intervention and streamlining operations [7]. Thirdly, it is important to note that the iterative nature of machine learning allows models to continuously improve and adapt to evolving datasets, resulting in enhanced accuracy and performance over time [8]. Despite its promising capabilities, machine learning for big data processing is not without challenges [9]. The main hurdle is that large data sets are complex to manage and effectively process [10]. Ensuring scalability, fault tolerance, and resource optimization in distributed environments remains an ongoing challenge. In addition, data quality [11] and preprocessing [12] are critical factors that can significantly impact the performance of machine learning models [13], [14]. Addressing issues such as missing values, outliers, and data imbalances requires careful consideration to prevent bias and inaccuracy in the analysis [15], [16]. Furthermore, the accuracy of individual machine learning models, which is critical for such tasks, is not always satisfactory. In particular, the method described [17] is highly effective for analyzing large datasets. However, the accuracy of their stochastic gradient descent (SGD) algorithm is not up to the mark.

To address this issue, Izonin *et al.* [18] investigated the nonlinear expansion of inputs for SGD, implementing it using different powers of the Wiener polynomial. This paper approximates a tabular dataset using different powers of the same polynomial. The degree of the polynomial was determined using SGD due to its high speed. The results of the modelling suggest that increasing the degree of approximation would improve the accuracy of the model. However, the direct approximation approach by high powers of this polynomial may significantly increase the problem's dimensionality. It is important to note that polynomial approximation may not always be appropriate when the number of attributes exceeds the number of vectors in a dataset. Hence, a direct approximation with this polynomial, even with the use of high-speed SGD, may not be the most suitable approach.

When discussing the composition of ensembles from these methods [19], it is important to note that they can partially alleviate the aforementioned issues. In particular, scaling, as the most accurate class of ensemble methods, can be optimized to work efficiently with large-scale datasets [20]. These methods partition the data or employ incremental training techniques, which can help manage and process data in distributed environments more efficiently. Cascade ensembles can be considered more fault-tolerant than individual models as they use multiple models [21]. This means that if one model fails or produces inaccurate results, the ensemble can still make reliable predictions by aggregating outputs from multiple models. Furthermore, cascade ensembles can optimize resources by distributing computation across multiple models or processing units [22], leading to optimized utilization of computational resources in distributed environments. Moreover, it is worth noting that ensemble methods, including cascade ensembles, have the potential to be resistant to noisy or imperfect data. By combining multiple models that are trained on different subsets or representations of the data, ensemble methods can mitigate the impact of missing values, outliers, and data imbalances [16].

Cascade ensemble methods may not completely solve all the challenges mentioned previously, but they can certainly help address them by utilizing the diversity and collective intelligence of multiple models [23]. It is crucial to carefully design and tune cascade ensembles to fit the specific characteristics and requirements of the problem domain [24]. While cascade ensembles are considered the most accurate class of ensemble methods, their hierarchical decision-making process requires a lengthy training procedure [25]. This task can become even more complex when analyzing high-dimensional datasets [26] using complex, nonlinear machine learning methods at each level of a deep learning cascade [27]. Moreover, the methodology entails segmenting the dataset into sections, which are then processed at designated levels within the cascade structure. This technique restricts the exposure of weaker predictors to the entire dataset, thereby reducing the accuracy of the cascade forecast or classification model as a whole [28]. These factors cumulatively affect the performance of the cascade ensemble. According to the literature, a comprehensive evaluation of the performance of cascade ensemble should take into account various indicators such as accuracy, speed, and generalization [29]. Accuracy (based on different performance indicators) measures how successfully the machine learning model predicts outcomes compared to the actual results. It's typically expressed as a percentage and is crucial for ensuring the reliability of insights derived from big data. Training time measures the duration required to train a machine learning model on a given dataset [30], while generalization measures its ability to perform well on unseen data. It is important to consider all these indicators in combination.

Dudzik *et al.* [27] developed a cascade ensemble based on support vector machines (SVMs). The SVM ensemble was composed using an evolutionary algorithm proposed by the authors to optimize the hyperparameters of the machine learning method underlying the cascade ensemble. The proposed approach has demonstrated high accuracy. The training process for SVMs [31] is known to have high time and memory complexity, which is further increased by the optimization procedure for each SVM at each level of the cascade. As a result, the accuracy and duration of SVMs are limited, making their application challenging. However, with careful consideration and expertise, SVMs can still be a valuable tool in certain contexts.

According to Izonin *et al.* [32], a distinct method was employed by the authors to construct a cascade ensemble using support vector regression (SVR). The dataset was partitioned into equal segments, with the number of segments determining the cascade's depth. The basic machine learning method used was linear SVR.

This approach improved the method's speed, although it may have decreased the potential accuracy of cascade ensembles.

To address the aforementioned drawback, a modification of this scheme was proposed in [33]. At each level of the cascade, the authors utilized high-speed SGD as a fundamental machine learning algorithm. Additionally, a quadratic Wiener polynomial was employed for the nonlinear transformation of input data at each level of the cascade, which resulted in a significant improvement in the forecast accuracy. Moreover, it should be noted that the cascade structure of the method results in an implicit approximation through a high-degree polynomial. It is worth mentioning that each new level of the cascade doubles the order of the Wiener polynomial. However, this increase in order leads to a significant expansion of the input data space, which in turn prolongs the training procedure. To partially compensate for this drawback, SGD is employed. However, it is worth noting that the proposed approach may have an impact on the generalization properties of the method. Therefore, it may be necessary to conduct further research to reduce the training time of the method while simultaneously improving its generalization properties and accuracy [34].

The objective of this paper is to improve the performance of the cascade ensemble of SGD classifiers by implementing a combination of a new data partitioning algorithm and PCA at each level of the ensemble method. The effectiveness of this approach is evaluated by measuring enhancements in the generalization properties and accuracy of the cascade ensemble of SGD classifiers, as well as a substantial reduction in the duration of its training. The main contributions of this paper are the following:

- We improved the SGD-based cascade ensemble by jointly utilizing a new data partitioning algorithm and additional application of PCA at each level of the hierarchical ensemble. The use of the first approach demonstrated a significant improvement in the accuracy of the ensemble method. The utilization of the second approach significantly reduced its training time. The combined use of both approaches provided a substantial enhancement in the performance of the cascade ensemble based on two critical indicators;
- We improved the training and application procedures of the cascade ensemble through the combined implementation of both approaches as mentioned in the first scientific contribution of this paper, improving its performance in terms of accuracy and training time when solving classification tasks, particularly in the analysis of large datasets;
- We have demonstrated a significant enhancement in the performance of the cascade ensemble (training time, generalization properties) compared to other possible implementations.

The paper is structured as follows: in section 2, the enhancements made to the cascade ensemble method are explained, including the implementation of a novel training data partitioning algorithm and the integration of principal component analysis (PCA) at each level. The results obtained from the application of the improved cascade ensemble method are presented in section 3, and the implications of the findings are discussed. Finally, section 4 summarizes the key findings and contributions of the study.

2. AN IMPROVED CASCADE ENSEMBLE METHOD

The cascade ensemble improved in this paper is based on [33]. As previously mentioned, Izonin *et al.* [33] proposed a hierarchical classifier that uses a high-speed SGD quadratic Wiener polynomial for nonlinear transformation of the input data at each level of the cascade. The training dataset is divided into equal parts, and the number of parts determines the number of cascade levels. However, it should be noted that the existing method has two drawbacks. One is the formation of random subsamples of the same size (without repetitions) for each level of the cascade. This may result in weak regressors receiving only a small portion of the useful information for analysis, which could potentially reduce the accuracy of the cascade as a whole. Second, the use of a nonlinear expansion scheme for the problem inputs based on the quadratic Wiener polynomial represents another potential disadvantage. While this approach has been shown to improve the accuracy of linear classifiers, it also expands the space of task inputs, which can result in a significant increase in training time, especially when dealing with high-dimensional data of large volume. However, in this paper, we aim to address both of these drawbacks. A new data partitioning algorithm was used in conjunction with additional application of PCA at each level of the hierarchical ensemble. The first approach showed a significant improvement in the accuracy of the ensemble method, while the second approach significantly reduced its training time. The combined use of both approaches provided a substantial enhancement in the performance of the cascade ensemble based on two critical indicators.

Let us take a closer look at these two methods. In this paper, we propose an additional use of PCA to reduce the dimensionality of the input data space for each weak classifier of the hierarchical method. The basic cascade ensemble [33] utilises a quadratic Wiener polynomial at each level, which can increase the dimensionality of the input data space. This approach allows for a more efficient and effective implementation of the method. To automate this procedure, we used the method of calculating cumulative variance explained values. It has been determined that the number of principal components can be automatically selected to meet the user's specified percentage of variance explained. Recent numerical modelling results have demonstrated

that selecting a value of 95% guarantees that each weak classifier considers the fundamental information necessary for the analysis. Our modification significantly reduces the input data space of the problem by discarding less significant or noisy principal components. This reduction is by more than 10 times and helps to shorten the training duration for both each weak machine learning-based classifier and the improved cascade ensemble as a whole. We call the use of this procedure in the baseline method [33] modification 1 [35].

Both the basic [33] and the improved cascade ensemble in this paper require dividing a large dataset into parts to form a cascade structure. A new data partitioning algorithm is proposed in the paper for forming subsets at each level of the cascade. This is achieved by randomly selecting a number of vectors from the training set that corresponds to the user-defined size. Thus, the subsets for the improved methods can be either larger or smaller in size compared to the subsets for the original method [33]. Additionally, the subsets may contain repeated vectors, which is not allowed in the existing method [33]. These modifications aim to enhance the accuracy of each weak classifier and the improved SGD-based cascade ensemble as a whole. The approach used in the original method [33] is referred to as modification 2 [36]. The performance of the existing cascade ensemble [33] can be enhanced by combining modification 1 and modification 2. However, it is necessary to modify the training algorithms and apply them to the improved SGD-based cascade ensemble. Please refer to Figure 1 for the flow chart of the improved SGD-based cascade ensemble training.

Let us explore the key stages of the enhanced training process and the application of the improved method using modification 1 and modification 2 in greater detail. In order to do so, we will introduce the concept of data processing procedure, which is utilized at every level of the improved SGD-based cascade ensemble, and outline its key stages. The data processing procedure consists of the following steps: i) normalization of data by columns based on the maximum element; ii) quadratic expansion of the normalized inputs of a given data sample via the wiener polynomial; and iii) applying PCA and selecting the number of principal components that provide 95% of the explained variance. Before beginning the training procedure, the training dataset is divided into subsets using a new data partitioning algorithm. This creates N-subsets with repetitions, which determine the N levels of the improved SGD-based cascade ensemble.

2.1. Learning algorithm

- Step 1. The data processing procedure is performed on the first subset to train the weak classifier 1 (SGD-based classifier 1).
- Step 2. The data processing procedure is performed on the second subset and applied to the pre-trained weak classifier 1. The output values obtained from SGD-based classifier 1 are added to subset 2 as an additional feature. After performing the data processing procedure, weak classifier 2 (SGD-based classifier 2) is trained.
- Step 3. The data processing procedure is performed on the third subset and applies it to the previously trained weak classifier 1. The output values obtained from SGD-based classifier 1 are added to subset 3 as an additional feature. Then, perform the data processing procedure and apply it to the pre-trained weak classifier 2. The output values obtained from SGD-based classifier 2 are added to subset 3 as an additional feature. Finally, after performing the data processing procedure, weak classifier 3 (SGD-based classifier 3) is trained.
- Step N. The data processing procedure is performed on the last, subset N and its application to the previously trained weak classifier 1. The output values obtained from SGD-based classifier 1 are added to subset N as an additional feature. Then, perform the data processing procedure and apply it to the pre-trained weak classifier 2. The output values obtained from SGD-based classifier 2 are added to subset N as an additional feature. Then, perform the data processing procedure and apply it to the pre-trained weak classifier 3. These steps are repeated at each subsequent level until the final level of the improved SGD-based cascade ensemble is reached, where the last weak classifier (SGD-based classifier N) is trained.

2.2. Application algorithm

In the application algorithm for the improved SGD-based cascade ensemble, the input data vector is classified into one of the problem-defined classes using a pre-trained cascade scheme with N levels. The given vector undergoes a data processing procedure and is then applied to the pre-trained weak classifier 1. The output values obtained from SGD-based classifier 1 were added to the given vector as an additional feature. Then, the data processing procedure was performed and the vector was applied to the pre-trained weak classifier 2. Subsequently, the output values obtained from SGD-based classifier 2 were added to the given data vector as another additional feature. The data processing procedure was performed again and the vector was applied to the pre-trained weak classifier 3.

All the steps outlined previously are carried out at each subsequent level until the state level of the improved SGD-based cascade ensemble is reached. At this point, the final weak classifier (SGD-based classifier N) is applied to determine the desired membership class of the input data vector. The improved cascade ensemble offers the following advantages:

- Improves the generalization properties of the data classification method;
- Increases the classification accuracy;
- Reduces subsample dimensionality at each cascade ensemble level;
- Reduces the complexity of computation of the selected weak classifier;
- Shortens the training procedure duration.

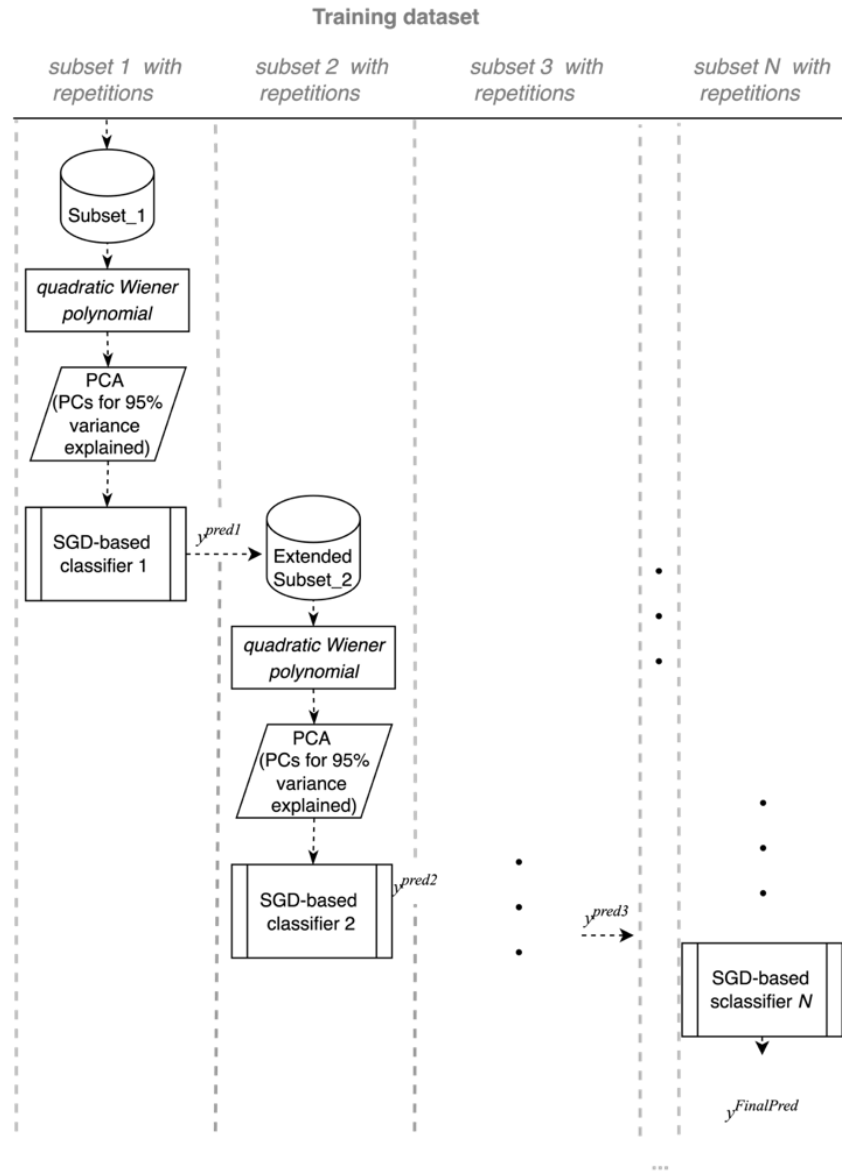


Figure 1. Flow-chart for the improved SGD-based cascade ensemble (training mode)

3. RESULTS AND DISCUSSION

To simulate the operation of the improved cascade ensemble, the authors created custom software using the Python programming language, based on principles from [37], [38]. Experimental studies were carried out on a computer with an Intel® Core™ i7-8750H processor (clock frequency 2.20 GHz), RAM 8 GB.

3.1. Dataset descriptions

The 2021 United States Disease Risk Factor Surveillance System (BRFSS) provided extensive datasets, which were disseminated by the Centers for Disease Control and Prevention across the United States and its surveyed regions [37]. The 2021 cycle of the BRFSS investigated a range of health parameters, such as overall health assessment, duration of wellness, physical activity levels, hypertension and cholesterol

screening, prevalence of chronic illness and arthritic conditions, tobacco usage patterns, fruit and vegetable consumption habits, and accessibility to medical assistance (core section). The primary dataset initially contained a broad spectrum of information related to these health conditions. This comprehensive dataset provides a solid foundation for further analysis and research. The dataset was refined to focus solely on lifestyle factors relevant to human health. Our main objective was to predict occurrences of cardiovascular diseases through a binary classification task. The resulting dataset consists of 308,854 records, each with 29 attributes.

The first phase of data preprocessing involved identifying and removing duplicate entries to ensure the uniqueness of each record. This step is crucial for maintaining the integrity of the dataset and preventing redundancy in the analysis. By eliminating duplicate entries, the researchers aimed to streamline the dataset and prepare it for further analysis. After ensuring the uniqueness of each record, the next step was to address and rectify any missing values within the dataset. Missing values can significantly impact the quality and reliability of the analysis. By addressing and rectifying these missing values, the researchers aimed to enhance the accuracy and robustness of the dataset for subsequent analytical processes. Following the preprocessing stage, the subsequent analytical step focused on mitigating the class imbalance observed in the dataset. Initially, the distribution ratio between classes stood at 92% to 8%. Maintaining balanced classes is crucial for the efficacy of machine learning models, as it can significantly impact the model's ability to make accurate predictions.

To achieve balanced classes, two principal algorithms were employed concurrently: synthetic minority oversampling technique (SMOTE) and NearMiss. SMOTE was used to augment instances of the minority class, while NearMiss was employed to reduce instances of the majority class. This iterative process involved adjusting various parameter values to regulate the number of instances from both classes, aiming to achieve a more balanced distribution. After a thorough iterative process of adjusting parameter values, it has been determined that the optimal accuracy and superior generalization for the used classifier can be achieved by selecting exactly 75,000 instances from each class in the original dataset. As a result, the final dataset for implementing machine learning training procedures contains 150,000 observations.

3.2. Optimal parameters selection for the improved cascade ensemble

The selection of parameters is of utmost importance for the improved SGD-based cascade ensemble. This ensemble jointly employs a new data partitioning algorithm and additional application of PCA at each level of the hierarchical ensemble. It is crucial to determine the following:

- The optimal number of levels for the cascade ensemble;
- The optimal size (% of the training sample) of each subset was randomly generated with repetitions of the improved cascade ensemble according to the enhanced training data separation algorithm;
- The optimal number of principal components at each level of the ensemble after applying PCA;
- Optimal parameters for each of the weak classifiers.

The optimal parameters of SGD were selected using the grid search method as a weak predictor at each level of the improved cascade ensemble. The number of principal components used in the hierarchical method was determined based on the cumulative variance explained. This approach allows for the calculation of the total variation in the data explained by a chosen number of principal components. For the optimization of the improved SGD-based cascade ensemble, we selected the principal components at each level of the cascade that accounted for 95% of the explained variance. These components were then used as inputs for each weak classifier. This approach ensured that the required number of principal components for each weak predictor of the ensemble was automatically selected, resulting in optimized performance. The implementation of automation in this procedure significantly reduces the time required to conduct research on the effectiveness of the method and its practical application.

In this paper, several experimental studies were conducted to determine the most efficient values for the first two parameters in order to optimize the effectiveness of the improved SGD-based cascade ensemble. This article presents the results of experiments on the accuracy, speed, and generalization properties of the proposed hierarchical method. The number of cascade levels varied from 2 to 6, and random subsamples with repetitions were formed at each level of the cascade ensemble of different sizes (ranging from 20% to 90% of the initial training sample with a step of 10%). The results are shown in Figure 2(a) shows the results in training mode, and Figure 2(b) shows the results in test mode.

Figure 2 demonstrates that utilizing a small percentage (20%-30%) of the training sample to create random subsamples with repetitions yields high accuracy during the training mode of the cascade ensemble. However, this approach may negatively impact its generalization properties. Conversely, using random subsamples with a volume of more than 50% of the training sample enhances accuracy during the application mode but may result in overtraining of the method. All of these characteristics apply to cascade designs with 2, 3, 4, 5, and 6 levels. It is important to note that using large subsamples at each level of the cascade ensemble may increase the training time of the entire method.

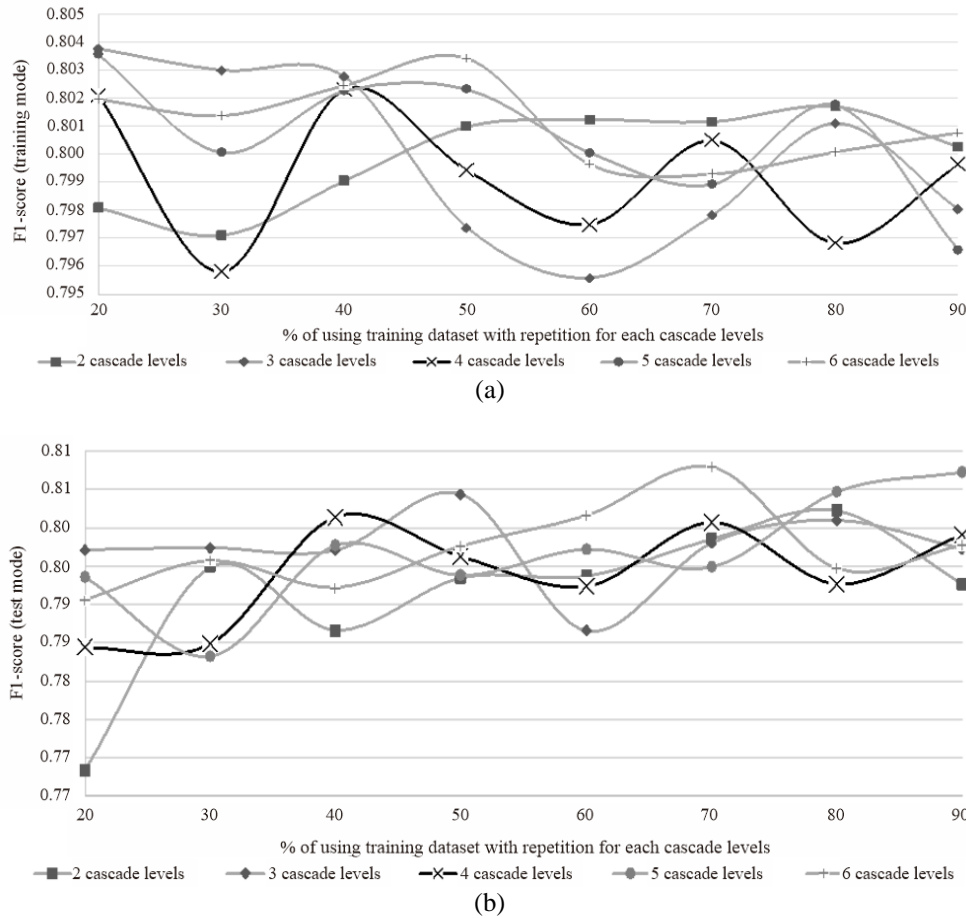


Figure 2. Change in the accuracy of the improved SGD-based cascade ensemble (F1-score) when changing the % of training sample usage with repetition and the number of ensemble levels: (a) for training mode and (b) for test mode

Figure 2 also illustrates that increasing the number of levels in the cascade ensemble (beyond 5) for processing a given dataset is not recommended, as it may reduce the generalization properties of the proposed design. We suggest that the optimal parameters for solving the problem are to use a cascade of four levels and to form random subsamples at each level using 40% of the training sample. These results are summarized in Figure 3. Based on the results present in Figure 3, it can be stated that the SGD-based cascade ensemble, when trained on subsamples comprising 40% of the training sample size, exhibits the highest generalization properties and accuracy in both modes. It is worth noting that the use of subsamples of this size does not significantly increase the training time of the method compared to the use of larger subsamples.

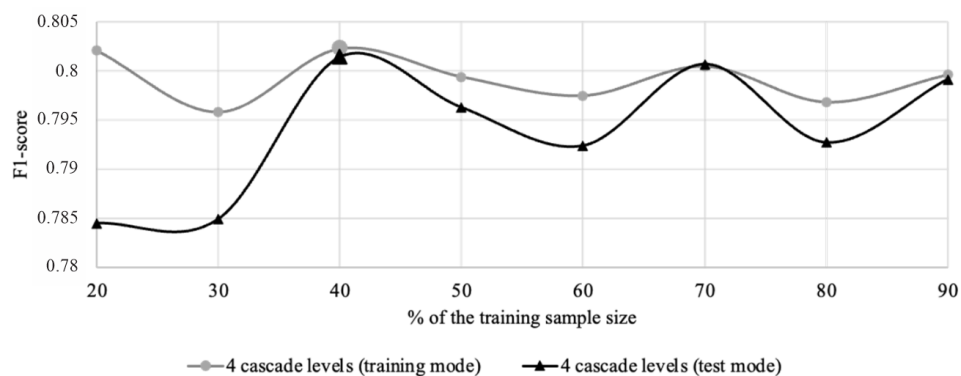


Figure 3. The best parameters for the improved SGD-based cascade ensemble

3.3. Results

Table 1 presents the results of the improved SGD-based cascade ensemble using four performance indicators. The results from Table 1 include the optimized parameters of the SGD-based cascade ensemble improved in this article. The results clearly demonstrate its superior performance compared to other methods.

Table 1. Results for the improved cascade ensemble

| Performance indicator | Training mode | Test mode |
|-------------------------|---------------|-----------|
| Precision | 0.806 | 0.884 |
| Recall | 0.803 | 0.75 |
| F1-score | 0.802 | 0.801 |
| Training time (seconds) | 0.31 | - |

3.4. Comparison and discussion

The proposed solution, the improved cascade ensemble, was evaluated for its effectiveness by comparing it with several similar methods: i) method [17] (SGD algorithm); ii) method [18] (extended-input SGD); iii) method [33] (basic cascade ensemble); iv) modification 1 (basic cascade ensemble with PCA on each cascade level) [35]; and v) modification 2 (basic cascade ensemble with subsamples with repetitions on each cascade level) [36]. The authors thoroughly analyzed the data set studied in this article and confidently investigated the effectiveness of each method mentioned above. They expertly selected the optimal parameters of each method using the grid search method. The results are summarized in Table 2.

Table 2. Optimal parameters for all investigated methods

| Method | Optimal parameters |
|------------------------|---|
| Method [17] | loss='log', penalty='l2', alpha=0.0001. |
| Method [18] | SGD classifier, quadratics Wiener polynomial. |
| Method [33] | Cascade ensemble of the SGD algorithms; quadratics Wiener polynomial; 5 depth levels; training sample is divided into 5 equal parts. |
| Modification 1 [35] | Cascade ensemble of the SGD algorithms; quadratics Wiener polynomial; 6 depth levels; input data space is reduced in dimensionality at each level of the cascade using PCA, ensuring at least 95% of the variance. |
| Modification 2 [36] | Cascade ensemble of the SGD algorithms; quadratics Wiener polynomial; 3 depth levels; training sample was subsampled randomly at each level, with 70% of the sample being selected with repetitions. |
| Proposed solution | Cascade ensemble of the SGD algorithms; quadratics Wiener polynomial; 4 depth levels; training sample was subsampled randomly at each level, with 40% of the sample being selected with repetitions; input data space is reduced in dimensionality at each level of the cascade using PCA, ensuring at least 95% of the variance. |

Two criteria were selected to compare the effectiveness of all the methods under study: i) F1-score in training and test modes; and ii) training time (in seconds); the first criterion provides an opportunity to compare the accuracy of all methods in application mode. Furthermore, the generalization properties of each method can be evaluated by the difference in F1 scores between training and test mode. The second criterion allows us to estimate the duration of the training procedure for the selected method, which is crucial when analyzing large datasets.

Figures 4 and 5 show the comparison results for F1 score and training time, respectively, based on both criteria. After careful analysis of the comparison results presented in Figures 4 and 5, it is clear that the method [17], which is based on the high-speed SGD algorithm, has the fastest training time in Figure 5, but exhibits the lowest classification accuracy as shown in Figure 4. However, this method still demonstrates high generalization properties. The method [18] was able to increase the classification accuracy of the data by more than 5% according to the F1-score in Figure 4, through the combination of the high-speed SGD algorithm and the quadratic Wiener polynomial. However, the use of quadratic Wiener polynomial significantly increases the dimensionality of the problem, which in turn leads to a longer training procedure. The training time for this method has increased significantly compared to the previous method see Figure 5. Furthermore, the

generalization properties of this method have deteriorated. Specifically, the difference between the F1-score in both training modes is significantly higher than that of method [17].

The basic cascade ensemble (method [33]) has been shown to achieve a 10% higher accuracy (F1-score) compared to method [17] and almost 5% compared to method [18]. Furthermore, this method has the advantage of reducing the duration of the training procedure by almost half compared to method [18]. But the basic cascade ensemble has the worst generalization properties of all the methods investigated as shown in Figure 4. The basic cascade ensemble (method [33]) was optimized by implementing PCA at each level (modification 1), resulting in a reduction of training time by almost 7 times. Furthermore, modification 1 led to a 2% increase in accuracy in the application mode. These benefits are attributed to the substantial reduction in subsample dimensionality at each level of the cascade ensemble. The number of problematic inputs at each level of the cascade ensemble was effectively reduced by more than tenfold by using PCA, which accounts for 95% of the variance. These least significant principal components either do not influence the classification results or are noise components negatively affecting the classification results. The accuracy of the basic cascade ensemble (method [33]) was improved by implementing a new subsampling scheme for each cascade level (modification 2). This resulted in a 2% increase in accuracy based on F1-score as shown in Figure 4. Additionally, this modification significantly enhanced the generalization properties compared to the method [33]. Nonetheless, the increased subsample size at each level of the cascade ensemble led to a training procedure that was nearly twice as long, as illustrated in Figure 5.

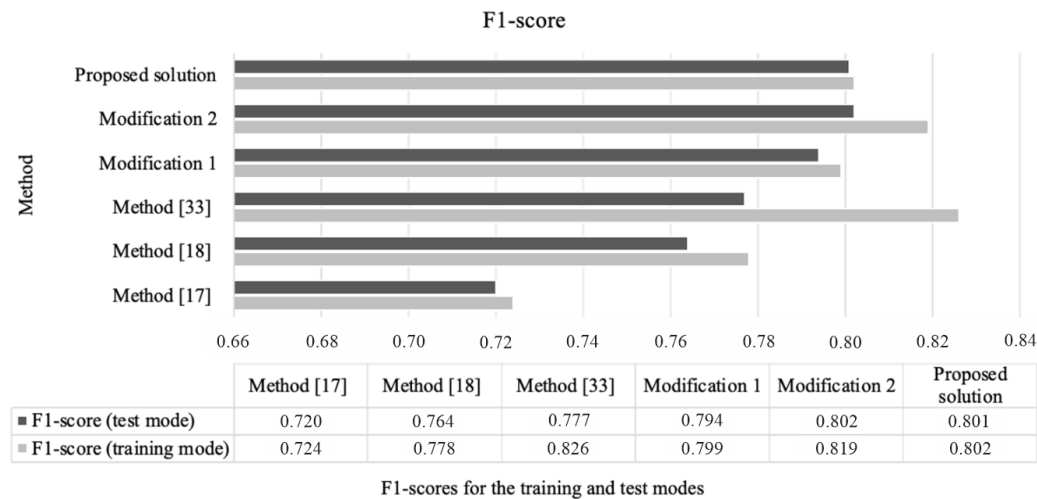


Figure 4. F1-score for all investigated methods

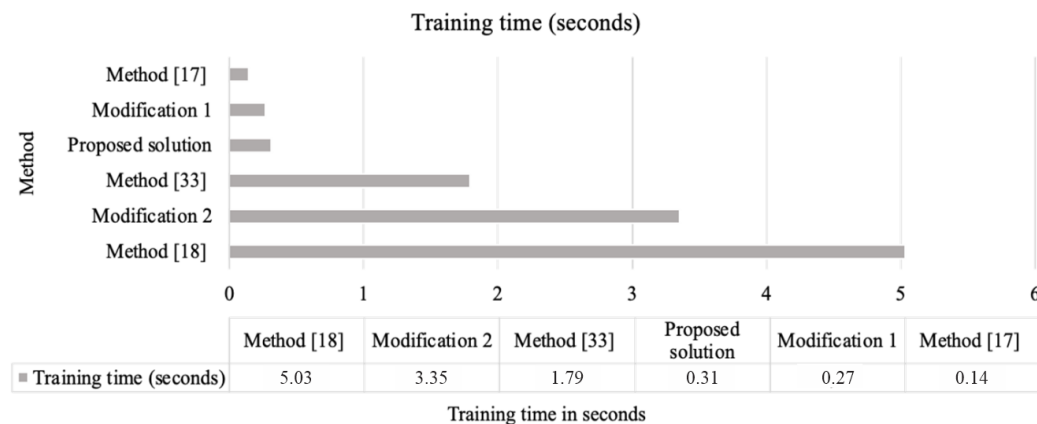


Figure 5. Training time (in seconds) for all investigated methods

The combined use of the new subsampling scheme and the application of PCA at each level of the basic cascade ensemble (method [33]), which is proposed in this article (proposed solution), provided high accuracy, the highest generalization, and significantly lower training time compared to the basic cascade

ensemble as shown in Figures 4 and 5. These advantages make this method a very practical solution. Future research could be directed towards two main areas:

- The accuracy of the improved cascade ensemble can be enhanced by using alternative linear methods as weak classifiers. It is important to consider this option only if these methods have been proven to provide higher accuracy than SGD when analyzing a specific dataset.
- Implementing alternative principal component extraction methods, such as neural network analogues of PCA, could potentially reduce the duration of the training procedure for the entire cascade ensemble, provided that they have a faster performance than the classical PCA.

However, both of the above approaches should be used taking into account the specifics of a particular task and the quality, quantity and dimensionality of the training data set available to solve it with machine learning.

4. CONCLUSION

The increase in digital data presents both challenges and opportunities. However, with the vast volume and complexity of big data, traditional processing methods can be strained. Machine learning is a key solution that enables analysis and insight extraction from large datasets. The integration of machine learning and big data has evolved significantly, with cascade ensembles, particularly ensemble methods, showing promise. When designing cascade ensembles, it is crucial to balance the factors of high accuracy and lengthy training, especially with complex datasets and nonlinear techniques. However, with careful consideration and expertise, cascade ensembles can still achieve impressive accuracy. Additionally, partitioning datasets into subsets can limit machine learning algorithms' access to the entire dataset, potentially affecting the overall performance of the cascade model. This paper presents significant improvements to the SGD-based cascade ensemble by integrating a new training data partitioning algorithm and PCA at each level. The combined use of these methods enhances the ensemble's accuracy and reduces training time. The paper demonstrates through modeling that the cascade ensemble's performance metrics, including accuracy, training time, and generalization properties, are significantly improved compared to the baseline method. Future research will explore alternative methodologies, such as non-iterative artificial neural networks (successive geometric transformations model (SGTM) neural-like structure) and neural network-based variations of PCA, to enhance accuracy, reduce training time, and maintain generalization properties with confidence. Furthermore, the examination of the cascade ensemble's presentation as a polynomial scheme (when utilizing SGTM neural-like structure as a weak classifier for the cascade) is intended to accelerate inference time during application stages. These inquiries have the potential to improve the capabilities of cascade ensembles and broaden their applicability in real-world big data scenarios.

ACKNOWLEDGEMENTS

Prof. Michal Gregus was supported by the Slovak Research and Development Agency under the contract No. APVV 19-0581. This work is funded by the European Union's Horizon Europe research and innovation program under grant agreement No 101138678, project ZEBAI (Innovative methodologies for the design of Zero-Emission and cost-effective Buildings enhanced by Artificial Intelligence).





REFERENCES

- [1] A. Chahal, P. Gulia, and N. S. Gill, "Different analytical frameworks and bigdata model for internet of things," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 25, no. 2, pp. 1159–1166, 2022, doi: 10.11591/ijeecs.v25.i2.pp1159-1166.
- [2] I. Krak, O. Stelia, A. Pashko, M. Efremov, and O. Khorozov, "Electrocardiogram classification using wavelet transformations," in *Proceedings - 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2020*, 2020, pp. 930–933, doi: 10.1109/TCSET49122.2020.235573.
- [3] L. Mochurad and N. Kryvinska, "Parallelization of finding the current coordinates of the lidar based on the genetic algorithm and openmp technology," *Symmetry*, vol. 13, no. 4, 2021, doi: 10.3390/sym13040666.
- [4] A. Chaudhary, K. R. Batwada, N. Mittal, and E. S. Pilli, "AdMap: a framework for advertising using MapReduce pipeline," *Computer Science and Information Technologies*, vol. 3, no. 2, pp. 82–93, 2022, doi: 10.11591/cs.it.v3i2.pp82-93.
- [5] A. H. Al-Hamami and A. A. Flayyih, "Enhancing big data analysis by using map-reduce technique," *Bulletin of Electrical Engineering and Informatics*, vol. 7, no. 1, pp. 113–116, 2018, doi: 10.11591/eei.v7i1.895.
- [6] Y. Marzhan, K. Talshyn, K. Kairat, B. Saule, A. Karlygash, and O. Yerbol, "Smart technologies of the risk-management and decision-making systems in a fuzzy data environment," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 28, no. 3, pp. 1463–1474, 2022, doi: 10.11591/ijeecs.v28.i3.pp1463-1474.
- [7] C. Ganguli, S. K. Shandilya, M. Nehrey, and M. Havryliuk, "Adaptive artificial bee colony algorithm for nature-inspired cyber defense," *Systems*, vol. 11, no. 1, 2023, doi: 10.3390/systems11010027.
- [8] L. Mochurad, K. Shakhovska, and S. Montenegro, "Parallel solving of fredholm integral equations of the first kind by Tikhonov regularization method using OpenMP technology," *Advances in Intelligent Systems and Computing*, pp. 25–35, 2020, doi: 10.1007/978-3-030-33695-0_3.
- [9] O. Bisikalo, V. Kharchenko, V. Kovtun, I. Krak, and S. Pavlov, "Parameterization of the stochastic model for evaluating variable small data in the Shannon entropy basis," *Entropy*, vol. 25, no. 2, 2023, doi: 10.3390/e25020184.





- [10] L. Rabhi, N. Falihi, L. Afraites, and B. Bouikhalene, "A functional framework based on big data analytics for smart farming," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 3, pp. 1772–1779, 2021, doi: 10.11591/ijeecs.v24.i3.pp1772-1779.
- [11] W. Maryati, I. O. Rahayuningrum, and A. I. Justika, "Quality of medical information determine the quality of diagnosis code," *International Journal of Public Health Science*, vol. 8, no. 3, pp. 326–331, 2019, doi: 10.11591/ijphs.v8i3.20236.
- [12] R. O. Boadu, P. Agyei-Baffour, and A. K. Edusei, "Data accuracy and completeness of monthly midwifery returns indicators of Ejisu Juaben Health Directorate of Ghana," *International Journal of Public Health Science*, vol. 8, no. 1, pp. 106–117, 2019, doi: 10.11591/ijphs.v8i1.15934.
- [13] O. Berezsky, O. Pitsun, P. Liashchynskyi, B. Derysh, and N. Batryn, "Computational intelligence in medicine," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 149, pp. 488–510, 2023, doi: 10.1007/978-3-031-16203-9_28.
- [14] O. Berezsky et al., "Fuzzy system for breast disease diagnosing based on image analysis," in *CEUR Workshop Proceedings*, 2019, pp. 69–83.
- [15] N. Bangera, Kayarvizhy, S. Luharuka, and A. S. Manek, "Improving time efficiency in big data through progressive sampling-based classification model," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 248–260, 2024, doi: 10.11591/ijeecs.v33.i1.pp248-260.
- [16] S. S. Kaddi and M. M. Patil, "Ensemble learning based health care claim fraud detection in an imbalance data environment," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 32, no. 3, pp. 1686–1694, 2023, doi: 10.11591/ijeecs.v32.i3.pp1686-1694.
- [17] Scikit learn, "SGDRegressor," *Scikit Learn*. Accessed: Feb. 08, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDRegressor.html
- [18] I. Izonin, M. Greguš ml, R. Tkachenko, M. Logoyda, O. Mishchuk, and Y. Kynash, "SGD-based wiener polynomial approximation for missing data recovery in air pollution monitoring dataset," *Advances in Computational Intelligence*, pp. 781–793, 2019, doi: 10.1007/978-3-030-20521-8_64.
- [19] S. C. Shivaprasad, P. P. Maruthi, T. S. Venkatesh, and V. K. Rajuk, "Ensemble model for accuracy prediction of protein secondary structure," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 32, no. 3, pp. 1664–1677, 2023, doi: 10.11591/ijeecs.v32.i3.pp1664-1677.
- [20] O. Mulesa, F. Geche, A. Batyuk, and V. Buchok, "Development of combined information technology for time series prediction," *Advances in Intelligent Systems and Computing*, vol. 689, pp. 361–373, 2018, doi: 10.1007/978-3-319-70581-1_26.
- [21] N. H. A. Malek, W. F. W. Yaacob, Y. B. Wah, S. A. Md Nasir, N. Shaadan, and S. W. Indratno, "Comparison of ensemble hybrid sampling with bagging and boosting machine learning approach for imbalanced data," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 29, no. 1, pp. 598–608, 2023, doi: 10.11591/ijeecs.v29.i1.pp598-608.
- [22] A. J. Barid, Hadiyanto, and A. Wibowo, "Optimization of the algorithms use ensemble and synthetic minority oversampling technique for air quality classification," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 3, pp. 1632–1640, 2024, doi: 10.11591/ijeecs.v33.i3.pp1632-1640.
- [23] D. P. Javale and S. S. Desai, "Machine learning ensemble approach for healthcare data analytics," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 28, no. 2, pp. 926–933, 2022, doi: 10.11591/ijeecs.v28.i2.pp926-933.
- [24] K. Dissanayake and M. G. M. Johar, "Two-level boosting classifiers ensemble based on feature selection for heart disease prediction," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 32, no. 1, pp. 381–391, 2023, doi: 10.11591/ijeecs.v32.i1.pp381-391.
- [25] A. P. Bakshi and V. K. Shandilya, "CCNNPD: Design of a cascade convolutional neural network for improved plant disease detection," in *AIP Conference Proceedings*, 2023, doi: 10.1063/5.0181602.
- [26] N. Krishnadoss and L. K. Ramasamy, "A study on high dimensional big data using predictive data analytics model," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 1, pp. 174–182, 2023, doi: 10.11591/ijeecs.v30.i1.pp174-182.
- [27] W. Dudzik, J. Nalepa, and M. Kawulok, "Ensembles of evolutionarily-constructed support vector machine cascades," *Knowledge-Based Systems*, vol. 288, 2024, doi: 10.1016/j.knsys.2024.111490.
- [28] A. R. Thatipalli, P. Aravamudu, K. Kartheek, and A. Dennisan, "Exploring and comparing various machine and deep learning technique algorithms to detect domain generation algorithms of malicious variants," *Computer Science and Information Technologies*, vol. 3, no. 2, pp. 94–103, 2022, doi: 10.11591/csit.v3i2.p94-103.
- [29] K. Xie, Y. Hou, and X. Zhou, "Deep centroid: a general deep cascade classifier for biomedical omics data classification," *Bioinformatics*, vol. 40, no. 2, 2024, doi: 10.1093/bioinformatics/btae039.
- [30] L. Mochurad, Y. Hladun, Y. Zasoba, and M. Gregus, "An approach for opening doors with a mobile robot using machine learning methods," *Big Data and Cognitive Computing*, vol. 7, no. 2, 2023, doi: 10.3390/bdcc7020069.
- [31] K. Zub, P. Zhezhnych, and C. Strauss, "Two-stage PNN–SVM ensemble for higher education admission prediction," *Big Data and Cognitive Computing*, vol. 7, no. 2, 2023, doi: 10.3390/bdcc7020083.
- [32] I. Izonin, R. Tkachenko, O. Gurbych, M. Kovac, L. Rutkowski, and R. Holoven, "A non-linear SVR-based cascade model for improving prediction accuracy of biomedical data analysis," *Mathematical Biosciences and Engineering*, vol. 20, no. 7, pp. 13398–13414, 2023, doi: 10.3934/mbe.2023597.
- [33] I. Izonin, R. Tkachenko, R. Holoven, K. Yemets, M. Havryliuk, and S. K. Shandilya, "SGD-based cascade scheme for higher degrees wiener polynomial approximation of large biomedical datasets," *Machine Learning and Knowledge Extraction*, vol. 4, no. 4, pp. 1088–1106, 2022, doi: 10.3390/make4040055.
- [34] S. Chimphee and W. Chimphee, "Machine learning to improve the performance of anomaly-based network intrusion detection in big data," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 2, pp. 1106–1119, 2023, doi: 10.11591/ijeecs.v30.i2.pp1106-1119.
- [35] I. Izonin, R. Muzyka, R. Tkachenko, I. Dronyuk, K. Yemets, and S. A. Mitoulis, "A method for reducing training time of ML-based cascade scheme for large-volume data analysis," *Sensors*, vol. 24, no. 15, 2024, doi: 10.3390/s24154762.
- [36] I. Izonin, R. Muzyka, R. Tkachenko, M. Gregus, N. Kustra, and S. A. Mitoulis, "An approach toward improvement of ensemble method's accuracy for biomedical data classification," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 5, pp. 5949–5960, 2024, doi: 10.11591/ijece.v14i5.pp5949-5960.
- [37] V. Yakovyna and N. Shakhovska, "Software failure time series prediction with RBF, GRNN, and LSTM neural networks," *Procedia Computer Science*, vol. 207, pp. 837–847, 2022, doi: 10.1016/j.procs.2022.09.139.
- [38] A. Sambir, V. Yakovyna, and M. Seniv, "Recruiting software architecture using user generated data," in *2017 13th International Conference Perspective Technologies and Methods in MEMS Design, MEMSTECH 2017 - Proceedings*, 2017, pp. 161–163, doi: 10.1109/MEMSTECH.2017.7937557.

BIOGRAPHIES OF AUTHORS







Dr. Ivan Izonin     is an Associate Professor at the Department of Artificial Intelligence of Lviv Polytechnic National University, Ukraine, and also a Research Fellow at the Department of Civil Engineering, School of Engineering, University of Birmingham, UK. He received an M.Sc. degree in Computer science from Lviv Polytechnic National University, Ukraine in 2011, and an M.Sc. degree in economic cybernetics from the Ivan Franko National University of Lviv, Ukraine, in 2012. He received a Ph.D. degree in artificial intelligence from the Institute of Computer Science and Information Technologies of Lviv Polytechnic National University, Ukraine in 2016. His research interests include small biomedical data analysis, meta-learning, and ensemble methods where he is the author/co-author of over 150 research publications. He can be contacted at email: ivan.v.izonin@gmail.com or i.izonin@bham.ac.uk.







Roman Muzyka     is a Ph.D. student at the Department of Artificial Intelligence of Lviv Polytechnic National University, Ukraine, and works as a data engineer. He received his master of science (M.Sc.) degree in computer science from Lviv Polytechnic National University, Ukraine in 2023. His primary research interests lie in ensemble methods, cascades, machine learning, and data analysis. Additionally, he was a member of the Junior Academy of Sciences of Ukraine from 2016 to 2018, actively contributing to various scientific initiatives. He also participated in the international Erasmus+ projects, where he focused on robotics, programming, and mathematical analysis. He can be contacted at email: roman.muzyka.mknssh.2022@lpnu.ua.







Prof. Roman Tkachenko     is a Professor at the Department of Publishing Information Technologies of Lviv Polytechnic National University, Ukraine since 2002. From 2017-2020, he was also the head of this department. He received an M.Sc. in electrical engineering from Lviv Polytechnic National University, Ukraine in 1972, and a Ph.D. in metrology from the Lviv Polytechnic National University, Ukraine in 1984. He received his doctor of science degree in information technologies from the State Research Institute of Information Infrastructure, Lviv, Ukraine. His research interests are primarily in computational intelligence, high-speed neural-like systems, non-iterative machine learning algorithms, and ensemble learning. He is the author/co-author of over 200 research publications. He can be contacted at email: roman.tkachenk@gmail.com.







Prof. Michal Greguš     is a Professor of the Faculty of Management, Comenius University Bratislava, Bratislava, Slovak Republic. He finished his university studies with summa cumlaude and obtained his Ph.D. degree in the field of mathematical analysis at the Faculty of Mathematics and Physics at Comenius University in Bratislava. He has been working previously in the field of functional analysis and its applications. At present, his research interests are in management information systems, in modelling of economic processes and in business analytics. He can be contacted at email: Michal.Gregus@fm.uniba.sk.



Prof. Roman Korzh     is a professor of the of the Department of Social Communications and Information Activities and also Vice-Rector for Research and Pedagogical Work and Social Development, Lviv Polytechnic National University, Ukraine. He received a Ph.D. in telecommunications in 1990 and his doctor of science degree in information technologies in 2019. His research interests are primarily in academic virtual communities, information image of universities, and web communities. He is the author/co-author of over 200 research publications. He can be contacted at email: roman.o.korzh@lpnu.ua.



Kyrylo Yemets     is a Ph.D. student at the Department of Artificial Intelligence of Lviv Polytechnic National University, Ukraine, and a machine learning engineer. He received an M.Sc. degree in computer science from the National Technical University "Kharkiv Polytechnic Institute", Ukraine in 2021. His research interests include time series, natural language processing, transformers, and ensemble methods where he is the author/co-author of over 10 research publications. He can be contacted at email: kyrylo.v.yemets@lpnu.ua.